

RANCANG BANGUN SISTEM PENGENALAN WAJAH UNTUK IDENTIFIKASI MAHASISWA YANG MELAKUKAN PELANGGARAN MELALUI BUKTI FOTO MENGGUNAKAN METODE EIGENFACE (Studi Kasus : Politeknik Negeri Malang)

Mohammad Chilman Fachrudik¹, Cahya Rahmad², Budi Harijanto³

^{1,2,3}Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang
¹moh.chilman@gmail.com, ²cahya.rahmad@polinema.ac.id, ³budi.hijet@gmail.com

Abstrak

Proses Identifikasi pelanggaran dan pencatatan pelanggaran di Jurusan Teknologi Informasi Politeknik Negeri Malang masih dilakukan secara manual. Proses identifikasi pelanggaran yang masih manual memiliki kelemahan seperti memakan waktu yang lama untuk proses identifikasi dan pencatatan pelanggarannya, rekap data agak sulit, dan dapat dimanipulasi oleh orang – orang yang kurang bertanggung jawab. Salah satu cara untuk mengatasi masalah tersebut dapat menggunakan teknologi image processing yaitu face recognition dengan membangun Sistem Pengenalan Wajah untuk Identifikasi Mahasiswa yang Melakukan Pelanggaran melalui Bukti Foto. Aplikasi ini diharapkan dapat mengetahui informasi mahasiswa yang melakukan pelanggaran secara akurat, efektif, dan efisien. Aplikasi ini dibuat menggunakan metode Eigenface dengan menggunakan bahasa pemrograman PHP serta menggunakan framework CI. Hasil dari pengujian yang telah dilakukan, aplikasi dapat mendeteksi dan mengenali citra wajah mahasiswa yang melakukan pelanggaran dari foto yang telah dimasukkan pada aplikasi untuk diidentifikasi. Pada proses identifikasi wajah juga dipengaruhi oleh jarak wajah serta posisi wajah yang mempengaruhi keakuratan dalam proses pengenalan wajah.

Kata kunci : *Image processing, Face recognition, Eigenfaces.*

1. Pendahuluan

Pelacakan dan pengenalan wajah manusia merupakan salah satu bidang penelitian yang penting, dan dewasa ini banyak aplikasi yang dapat menerapkannya, baik di bidang komersial maupun bidang penegakan hukum. Penelitian terhadap pengenalan wajah manusia sudah banyak dilakukan dengan kelebihan dan kekurangan tertentu. Hal ini disebabkan karena wajah manusia mempresentasikan sesuatu yang kompleks, sehingga untuk mengembangkan model komputasi yang ideal untuk pengenalan wajah manusia adalah sesuatu hal yang sulit.

Pentingnya pengenalan wajah dalam kasus pelanggaran di Politeknik Negeri Malang adalah untuk memudahkan dalam proses identifikasi pelaku dari wajah yang tertangkap di sebuah foto saat pelaku melakukan sebuah pelanggaran. Sehingga informasi pelaku dapat diketahui dengan cepat dan jelas serta pelanggaran – pelanggaran apa saja yang pernah dilakukan oleh si pelaku dapat diketahui.

2. Tinjauan Pustaka

2.1 Biometrika

Sistem biometrika merupakan teknologi pengenalan diri dengan menggunakan bagian tubuh atau perilaku manusia. Sidik jari, tanda tangan, DNA, telinga, wajah, infrared, gaya berjalan, geometri tangan, selaput pelangi, gaya penekanan tombol, bau, telapak tangan, retina, suara, gigi, dan bibir. Masing-masing merupakan karakteristik dari cabang ilmu biometrika yang dewasa ini sedang berkembang pesat. Sistem pengenalan diri adalah sistem untuk mengenali identitas seseorang secara otomatis dengan menggunakan teknologi komputer, yang bertujuan untuk meningkatkan keamanan sistem dan untuk mengenali target secara cepat dan tepat.

2.2 Pengenalan Wajah

Wajah merupakan bagian dari tubuh yang berperan penting dalam proses penyampaian ciri, identitas dan emosi seseorang. Kemampuan manusia dalam mengenali wajah sering terjadi secara tidak sadar, manusia mampu mengenali ribuan wajah

sepanjang hidupnya dan mengidentifikasi wajah yang sekilas dikenalnya sampai beberapa tahun kemudian. Proses ini berlangsung begitu cepat dan dapat tersimpan cukup lama dalam memori manusia walaupun wajah yang dikenalnya memiliki banyak perubahan visual seperti adanya perubahan kondisi, ekspresi, sudut pandang, penuaan dan penambahan aksesoris seperti kacamata, topi sampai adanya perubahan gaya rambut. Oleh karena itu wajah digunakan sebagai acuan indikasi untuk mengenali seseorang.

2.3 Pengolahan Citra

Pengolahan citra merupakan suatu proses yang dilakukan dengan input berupa citra dan output nya pun berupa citra. Proses ini dilakukan pada citra data training sebelum proses feature extraction (pengambilan ciri). Terdapat dua hal mendasar dalam memahami proses pembentukan citra, yaitu

1. Geometri formasi citra yang menentukan lokasi suatu titik dalam pemandangan yang diproyeksikan pada bidang citra.
2. Fisik cahaya yang menentukan brightness suatu piksel citra sebagai fungsi pencahayaan.

Oleh karena itu, diperlukan penghubung antara notasi matematika untuk mengembangkan algoritma pengolahan citra dan notasi algoritma yang digunakan untuk pembuatan program komputer yang disimpan kedalam sistem penyimpanan memori dua dimensi yang disebut larik (array).

2.4 Eigenfaces

Pendekatan Eigenface adalah salah satu metode paling sederhana dan paling efisien untuk pengenalan wajah. Dalam pendekatan eigenface memilih nilai threshold yang merupakan faktor yang sangat penting bagi performace pengenalan wajah. Selain itu, pengurangan dimensi ruang wajah tergantung pada jumlah eigenfaces yang diambil. Penelitian diatas pernah diteliti oleh Gupta, et al. (2010). Kelebihan dari penelitian diatas adalah dapat meningkatkan performance pada pengenalan wajah ketika kedua faktor tersebut dikombinasikan yaitu 15% dari eigenfaces dengan nilai eigen terbesar dipilih dan nilai threshold yang dipilih 0,8 kali maksimum jarak Euclidean minimum setiap gambar dari semua gambar yang lain. Sedangkan kekurangan dari penelitian ini adalah dilakukan penambahan proses pada preprocessing yaitu threshold.

3. Metodologi

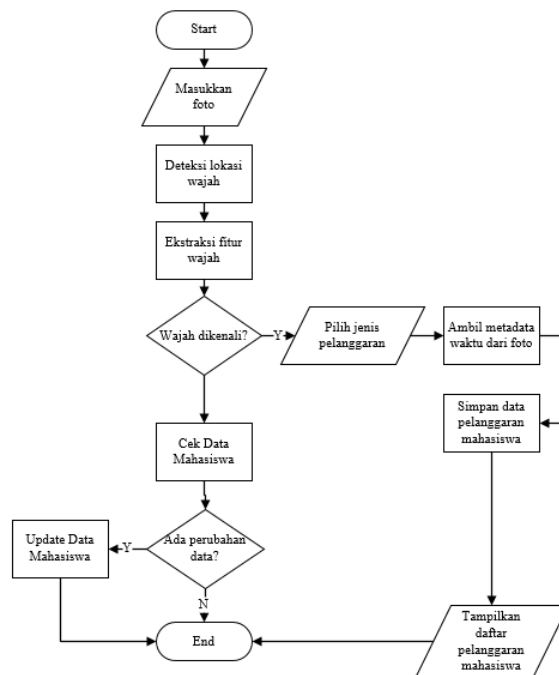
Pada penelitian skripsi ini menggunakan dua metode, yaitu metode studi literatur dan metode eksperimental. Metode studi literatur adalah tahap pencarian data mengenai pengenalan wajah, algoritma eigenface, serta penerapan pada sistem identifikasi wajah. Sedangkan pada metode

eksperimental adalah tahap untuk melakukan rancangan dan implementasi metode eigenface pada sistem identifikasi wajah serta dilakukan pengujian dan pengambilan data dari pengujian tersebut.

4. Perancangan dan Implementasi

4.1 Proses Identifikasi Wajah

Proses identifikasi wajah mahasiswa yang melakukan pelanggaran di perlukan beberapa tahap, pada proses pengenalan wajah dapat terlihat seperti pada gambar 1.



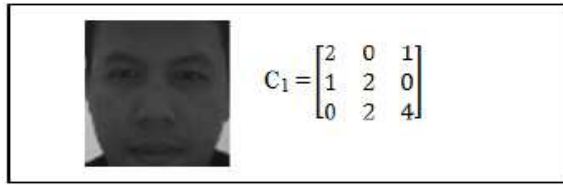
Gambar 1. Flowchart Identifikasi Pelanggaran

4.2 Perhitungan Eigenfaces

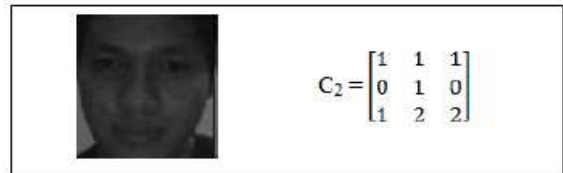
Algoritma pengenalan wajah dilakukan melalui beberapa tahapan, tahap pertama yaitu menyiapkan data dengan membuat suatu himpunan matriks yang ada di database, ambil nilai tengah atau mean, cari selisih antara *training image* dengan nilai tengah, hitung nilai matriks kovarian, menghitung *eigenvalue* dan *eigenvector*, tentukan nilai *eigenface*, dan terakhir adalah identifikasi. Untuk lebih jelasnya akan dijelaskan sebagai berikut.

4.2.1 Penyusunan Flatvector Matriks Citra

Langkah pertama adalah menyusun suatu himpunan S matriks yang terdiri dari seluruh *training image* (I_1, I_2, \dots, I_m). Misalnya, *training image* terdapat dua data wajah seperti terlihat pada gambar 2 dan 3, yang masing-masing memiliki nilai matriks.



Gambar 2. Citra wajah 1



Gambar 3. Citra wajah 2

4.2.2 Hitung Nilai Tengah atau Mean (Ψ)

Dari himpunan matriks yang telah diperoleh, langkah selanjutnya adalah mencari nilai tengah atau mean Ψ . Jumlahkan nilai matrik wajah 1 dan wajah 2 kemudian bagi dengan jumlah data wajah yang ada di database, pada penelitian ini jumlah data wajah yang ada di database adalah dua data wajah.

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n \quad (1)$$

Cari nilai tengah atau mean (Ψ).

$$\begin{aligned} \Psi &= \frac{1}{2} \sum_{n=1}^2 \Gamma_n = \frac{1}{2} \left[\begin{bmatrix} 2 & 0 & 1 \\ 1 & 2 & 0 \\ 0 & 2 & 4 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 2 & 2 \end{bmatrix} \right] \\ &= \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 2 & 3 \end{bmatrix} \end{aligned}$$

4.2.3 Hitung Selisih antara Training Image dengan Nilai Tengah

Dengan memakai nilai tengah citra di atas, langkah selanjutnya adalah mencari selisih (ϕ) antara training image (Γ) dengan nilai tengah (Ψ), dengan mengurangi training image (Γ) dengan nilai tengah (Ψ).

$$\begin{aligned} \phi_i &= \Gamma_i - \Psi \\ \phi_1 &= \Gamma_1 - \Psi = \begin{bmatrix} 2 & 0 & 1 \\ 1 & 2 & 0 \\ 0 & 2 & 4 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 2 & 3 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (2)$$

$$\begin{aligned} \phi_2 &= \Gamma_2 - \Psi = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 2 & 2 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 2 & 3 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix} \end{aligned}$$

4.2.4 Hitung Nilai Matriks Kovarian

Nilai matriks kovarian (C) digunakan untuk menghitung eigenvalue (λ) dan eigenvector (v).

$$C = \frac{1}{M} \sum_{n=1}^M \phi_n \phi_n^T = A A^T \quad (3)$$

$$L = A^T A \quad L = \theta_m^T \theta_n \quad (4)$$

Hitung nilai matriks kovarian (C)

$$\begin{aligned} L &= \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix} \end{aligned}$$

4.2.5 Hitung Nilai Eigenvalue dan Eigen-vector

Langkah selanjutnya adalah menghitung nilai eigenvalue (λ) dan eigenvector (v) dari matriks kovarian (C).

$$C x v_i = \lambda_i x v_i \quad (5)$$

Cari nilai nilai eigenvalue (λ) dan eigen-vector (v).

$$\begin{aligned} L x v &= \lambda x v \\ L x v &= \lambda I x v \end{aligned} \quad (6)$$

$$(L - \lambda I) = 0 \text{ atau } (\lambda I - L) = 0$$

$$\begin{aligned} 0 &= \lambda \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix} \\ 0 &= \det \begin{bmatrix} \lambda - 2 & -1 & 0 \\ -1 & \lambda - 2 & 0 \\ 0 & 0 & \lambda - 3 \end{bmatrix} \end{aligned}$$

Maka eigenvalue yang dihasilkan adalah $\lambda_1=3$, $\lambda_2=1$, $\lambda_3=3$

$$v = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

Eigenvector (v) dihasilkan dengan men-substitusikan nilai eigenvalue (λ) kedalam persamaan $\lambda I - L v = 0$. Eigenvector dari masing-masing eigenvalue didapat berdasarkan masing-

masing kolom *eigen-value* dan kemudian dihimpun kembali menjadi satu matriks.

- Untuk $\lambda_1 = 3$, $\lambda_2 = 1$, dan $\lambda_3 = 3$ maka :

$$\begin{bmatrix} \lambda - 2 & -1 & 0 \\ -1 & \lambda - 2 & 0 \\ 0 & 0 & \lambda - 3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Dihasilkan *eigenvector* v_1 adalah

$$\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$$

Dihasilkan *eigenvector* v_2 adalah

$$\begin{bmatrix} -1 \\ -1 \\ 0 \end{bmatrix}$$

Dihasilkan *eigenvector* v_3 adalah

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Setelah didapat *eigenvector* v_1, v_2 , dan v_3 , maka *eigenvector* yang dihasilkan dari matriks L adalah

$$\begin{bmatrix} 1 & -1 & 0 \\ -1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

4.2.7 Nilai Eigenface

Langkah selanjutnya, setelah *eigen-vector* (v) diperoleh, maka nilai *eigenface* (μ) dapat dicari dengan:

$$\mu_i = \sum_{k=1}^M v_{ik} \phi_k \quad (7)$$

$l = 1, \dots, M$

Cari nilai *eigenface* (μ) :

$$\begin{aligned} \mu_1 = v \times \phi_1 &= \begin{bmatrix} 1 & -1 & 0 \\ -1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & -1 & 0 \\ -2 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

$$\begin{aligned} \mu_2 = v \times \phi_2 &= \begin{bmatrix} 1 & -1 & 0 \\ -1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

4.2.8 Proses Identifikasi

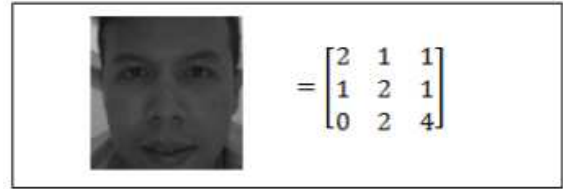
Proses identifikasi adalah proses dimana wajah baru masuk dan akan dicocokkan dengan wajah

yang ada di dalam database untuk mengetahui wajah mana yang cocok antara wajah yang baru masuk dengan wajah yang ada di dalam database. Untuk mengenali wajah baru yang masuk (test face), langkah yang dilakukan sama dengan data wajah yang ada di database, untuk mendapatkan nilai *eigenface* dari wajah baru.

$$\begin{aligned} \mu_{new} &= v \times (\Gamma_{new} - \Psi) \\ \Omega &= [\mu_1, \mu_2, \dots, \mu_M] \end{aligned} \quad (8)$$

Pertama cari selisih (Φ) antara test face dengan nilai tengah (Ψ). nilai matriks test face dari koordinat tersebut terlihat pada gambar 4.

$$\Phi_{new} = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 0 & 2 & 4 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$



Gambar 4. Citra wajah baru (*testface*)

Selanjutnya, setelah selisih (Φ) antara *testface* dengan nilai tengah (Ψ) di-dapat, maka nilai *eigenface* dapat dicari.

$$\begin{aligned} \mu_{new} &= v \times \Phi_{new} \\ \mu_{new} &= \begin{bmatrix} 1 & -1 & 0 \\ -1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & -1 \\ -2 & -2 & -1 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned} \quad (9)$$

Gunakan *euclidean distance* untuk mencari selisih terkecil antara *eigenface training image* (Γ_i) dalam *database* dengan *eigenface test face* (Γ_{new}), setelah itu jumlahkan matriks dari masing-masing *euclidean distance*.

$$\begin{aligned} \varepsilon_k &= \|\Omega - \Omega_k\| \\ \varepsilon_k &= \|\Omega - \Omega_{new}\| \\ \varepsilon_1 &= \|\Omega - \Omega_{new}\| \\ &= \left\| \begin{bmatrix} 0 & 0 & -1 \\ -2 & -2 & -1 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & -1 & 0 \\ -2 & -2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right\| \\ &= \sqrt{(0)^2 + (1)^2 + (1)^2 + (0)^2 + (1)^2 + (1)^2 + (0)^2 + (0)^2 + (0)^2} \\ &= \sqrt{4} = 2 \end{aligned} \quad (10)$$

$$\begin{aligned}
 \varepsilon_2 &= \|\Omega - \Omega_{new}\| \\
 &= \left\| \begin{bmatrix} 0 & 0 & -1 \\ -2 & -2 & -1 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right\| \\
 &= \sqrt{(0)^2 + (1)^2 + (1)^2 + (4)^2 + (1)^2 + (1)^2 + (0)^2 + (0)^2 + (0)^2} \\
 &= \sqrt{8} = 2,828
 \end{aligned}$$

Dalam hasil perhitungan, diperoleh jarak *eigenface* citra wajah satu memiliki jarak yang terkecil. Karena jarak *eigenface* wajah satu dengan testface yang paling kecil, maka hasil identifikasi menyimpulkan bahwa testface lebih mirip dengan wajah satu dibandingkan dengan wajah dua.

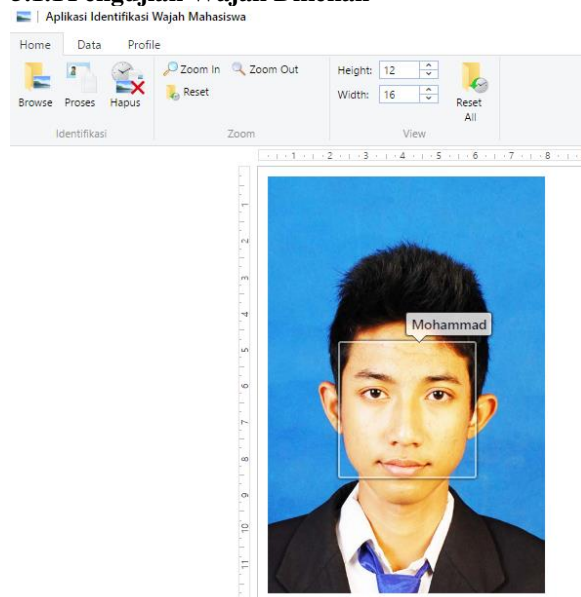
4.3 Perancangan Sistem

Perancangan sistem yang digunakan padapenelitian ini adalah DFD, ERD, *Class Diagram*, *Sequence Diagram*, rancangan antar mukaserta rancangan database.

5. Pengujian dan Analisa Hasil

5.1 Pengujian Pengenalan Wajah

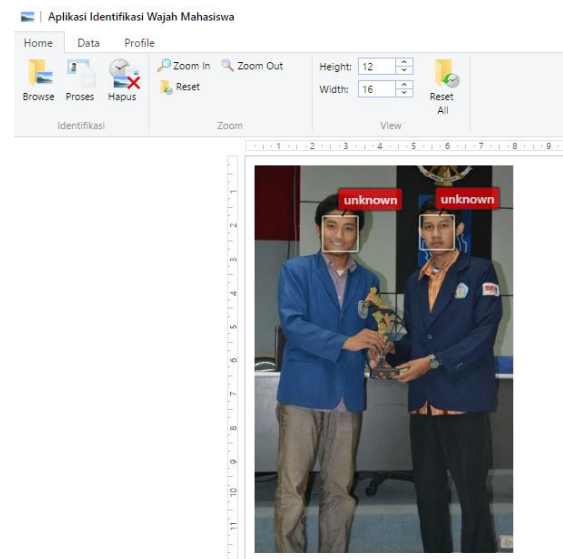
5.1.1 Pengujian Wajah Dikenali



Gambar 5. Identifikasi wajah dikenali

Apabila pada foto yang dimasukkan untuk identifikasi wajah dimasukkan foto wajah yang terdapat dalam sistem, maka wajah tersebut akan dikenali dan menampilkan nama pemilik wajah tersebut.

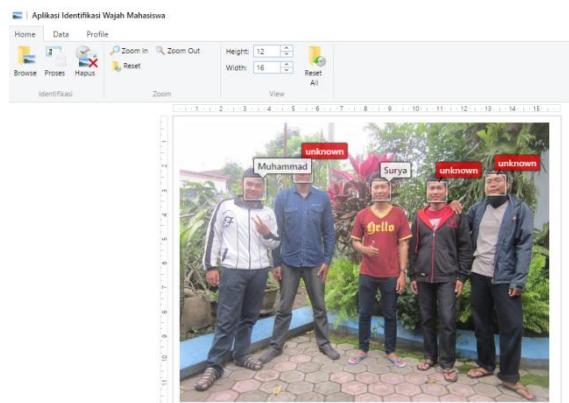
5.1.2 Pengujian Wajah Tidak Dikenali



Gambar 6. Identifikasi wajah tidak dikenali

Apabila pada foto yang dimasukkan untuk identifikasi wajah dimasukkan foto wajah yang tidak terdapat dalam sistem, maka wajah tersebut tidak akan dikenali dan menampilkan tulisan “unknown” yang artinya wajah tidak dikenali.

5.1.3 Pengujian Wajah Pada Jarak Tertentu



Gambar 7. Identifikasi wajah pada jarak tertentu

Apabila pada foto yang dimasukkan untuk identifikasi wajah dimasukkan foto wajah yang berjarak cukup jauh dari kamera dan juga terdapat beberapa wajah. Maka wajah yang dapat dikenali hanya wajah yang terdapat dalam sistem dan wajah harus tampak jelas sehingga sistem dapat mengenali wajah tersebut.

5.1.4 Pengujian Keberhasilan Metode

Dalam proses pengenalan wajah menggunakan metode *eigenface*, proses keberhasilannya

dipengaruhi oleh berbagai faktor, yaitu pencahayaan, jarak, dan juga aksesoris yang dipakai pemilik wajah. Dalam proses pengenalan wajah menggunakan metode *eigenface* ini waktu dalam proses pengenalan wajahnya juga dipengaruhi oleh jumlah data wajah yang ada dalam sistem. Semakin banyak data wajah pada sistem, semakin lama juga proses pengenalan wajahnya. Dalam proses pengenalan wajah juga dapat memberikan batasan nilai wajah yang tidak dapat dikenali oleh sistem. Oleh karena itu, diperlukan batas dari nilai pengenalan wajah yang disebut juga *Threshold*. Dalam sistem ini nilai *Threshold* untuk membatasi dalam pengenalan wajahnya adalah sebesar 2,6. Apabila nilai minimal dari wajah yang didapat dari data sistem lebih dari 2,6, maka wajah tersebut dianggap tidak dikenali karena melebihi nilai *Threshold* yang telah ditentukan.

Dari proses pengenalan wajah dilakukan beberapa pengujian, yaitu:

- Proses pengenalan wajah berdasarkan jarak

$$= \frac{24}{30} \times 100 = 80\%$$

- Proses pengenalan wajah berdasarkan cahaya

$$= \frac{22}{30} \times 100 = 73\%$$

- Proses pengenalan wajah berdasarkan

$$\text{ekspresi wajah} = \frac{23}{30} \times 100 = 76\%$$

6. Kesimpulan

Berdasarkan hasil dari perancangan Sistem Pengenalan Wajah untuk Identifikasi Mahasiswa yang Melakukan Pelanggaran Melalui Bukti Foto ini, dapat diambil beberapa kesimpulan sebagai berikut:

- a. Pada proses pengenalan wajah perlu dilakukan perbandingan antara wajah yang telah tersimpan di database dengan wajah yang terdeteksi pada foto. Proses perbandingan tersebut menggunakan perhitungan metode *eigenface*. Setelah didapatkan hasil perhitungan, maka akan dicari nilai terkecil sebagai hasil dari wajah yang akan dikenali.
- b. Untuk dapat mengenali wajah dengan baik dan tidak salah pengenalan, maka perlu menentukan *threshold* yang sesuai agar dapat mengenali wajah. *Threshold* berfungsi sebagai batas nilai dari proses ekstraksi fitur wajah. Dari beberapa pengujian ditemukan nilai *threshold* yang sesuai adalah 2,6.
- c. Histori pelanggaran dapat dilihat dari data pelanggaran yang telah ditambahkan pada proses identifikasi wajah si pelaku pelanggaran. Sehingga user dapat melihat apa saja pelanggaran yang pernah dilakukan

oleh mahasiswa tersebut dari data pelanggaran yang telah tersimpan.

- d. Untuk menentukan waktu pelanggaran selain memasukkan tanggal sendiri, juga dapat dilakukan dengan mendapatkan tanggal dari bukti foto yang diidentifikasi. Tanggal tersebut dapat diperoleh dengan mendapatkan metadata *date* dari foto tersebut.

7. Saran

Dalam skripsi ini di ajukan beberapa saran untuk pengembangan Sistem Pengenalan Wajah untuk Identifikasi Mahasiswa yang Melakukan Pelanggaran Melalui Bukti Foto ini, yaitu:

- a. Aplikasi identifikasi mahasiswa ini bisa dikembangkan agar dapat melakukan pengenalan wajah lebih akurat lagi dan dapat melakukan pengenalan wajah dari berbagai arah.
- b. Kedepannya aplikasi ini dapat dikembangkan menjadi aplikasi berbasis mobile sehingga proses identifikasi bisa langsung dilakukan secara mudah.

Daftar Pustaka:

- Harjoseputro, Y., et all. 2015. *Perbandingan Waktu Dan Tingkat Akurasi Pada Pengenalan Wajah Dengan Dan Tanpa Menggunakan Dekomposisi Citra*. Yogyakarta: Universitas Atmajaya Yogyakarta
- Hermawati, F. A. 2013. *Pengolahan Citra Digital*. Yogyakarta: Penerbit Andi
- Indra. 2012. *Sistem Pengenalan Wajah Dengan Metode Eigenface Untuk Absensi Pada PT. florindo lestari*. Jakarta: Universitas Budi Luhur
- Kurniawan, A., Saleh A., & Ramadijanti N. 2013. *Aplikasi Absensi Kuliah Berbasis Identifikasi Wajah Menggunakan Metode Gabor Wavelet*. Surabaya: Institut Teknologi Sepuluh Nopember
- Muliawan, M.R., Irawan, B., & Brianorman, Y. 2015. *Implementas Pengenalan Wajah Dengan Metode Eigenface Pada Sistem Absensi*. [Online] Tersedia: <http://jurnal.untan.ac.id/index.php/jcskcommip/a/article/download/9727/9500> [6 Desember 2015]
- Pujihati, R. 2014. *Penerapan Metode Jaringan Syaraf Tiruan Learning Vector Quantization (lvq) Untuk Pengenalan Wajah Dengan Citra Wajah Gaussian Blur*. Bandung: UPI
- Rahman, M. A. 2010, "Sistem Pengenalan Wajah Menggunakan Webcam Untuk Absensi Dengan Metode Template Matching", Surabaya: Politeknik Elektronika Negeri Surabaya

- Sholeh, A. 2013. *Pengembangan Sistem Pengenalan Wajah 2d Dengan Implementasi Algoritma Eigenface Dan Manhattan Distance*. Bandung: UPI
- Suprianto, D. 2013. *Sistem Pengenalan Wajah Secara Real-Time, dengan Adaboost, Eigenface PCA & MySql*. Malang: Universitas Brawijaya Malang